

## **AMENDMENTS TO THE CLAIMS**

The following listing of claims will replace all prior versions and listings of claims in the application.

### **LISTING OF CLAIMS**

1. (Currently Amended) A context-aware tokenizer comprising:  
at least one context automaton module that generates a context record associated with ~~token~~text strings of an input data stream;  
a tokenizing automaton module having a token automaton that partitions said input data stream into substrings corresponding to predefined tokens based on pattern information contained in said token automaton while simultaneously verifying contextual appropriateness based on said context record by taking context of a token into account and using it as a precondition to its recognition.
2. (Currently Amended) The tokenizer of claim 1 wherein said context automaton module comprises a left context automaton that populates said context record based on identified patterns that precede a given ~~token~~text string and a right context automaton that populates said context record based on identified patterns that follow said given text string~~token~~.
3. (Original) The tokenizer of claim 1 wherein said tokenizing automaton module maintains a data store of predefined token classes, and assigns each token identified to at least one of said predefined token classes.

4. (Original) The tokenizer of claim 3 wherein said tokenizer reports information indicative of the position and class membership of tokens identified.

5. (Original) The tokenizer of claim 1 wherein said tokenizing automaton defines a failure state, and wherein said tokenizing automaton module monitors the occurrence of said failure state to maintain a record of the longest match found involving said failure state to detect a default token in the absence of any matching patterns taken from said context automaton module and said tokenizing automation module.

6. (Original) The tokenizer of claim 1 wherein said context automaton scans said input data stream in a left-to-right direction to acquire left context information and in a right-to-left direction to acquire right context information.

7. (Original) The tokenizer of claim 1 wherein said context automaton and said tokenizing automaton collectively obey a linear time operating constraint.

8. (Currently Amended) A text-to-speech synthesizer according to claim 1 wherein ~~said input data stream is a text string and~~ said tokenizing automaton module partitions said text strings to include token class membership information from which the pronunciation of said text strings by said synthesizer is influenced.

9. (Currently Amended) A text processor according to claim 1 wherein ~~said input data stream comprises text and~~ said tokenizing automaton is coupled to said text processor and operates upon said text strings to identify and label multi-word phrases for single unit treatment by said text processor based on information extracted by said context automaton.

10. (Currently Amended) A text processor according to claim 1 wherein said input data stream lacks word unit separation symbols and wherein said tokenizing automaton module is coupled to said text processor and operates upon said text strings to identify and label word units for single unit treatment by said text processor based on information extracted by said context and token automata.

11. (Currently Amended) A method of tokenizing an input stream comprising:  
using at least one context automaton to generate a context record associated with ~~token~~text strings of said input stream;

using at least one tokenizing automaton to segment said input stream into predefined substrings corresponding to tokens by taking context of a token into account and using it as a condition precedent to its recognition~~based on pattern information contained in said context record.~~

12. (Currently Amended) The method of claim 11 wherein said step of generating said context record is performed using a left context automaton to populate said context record based on identified patterns that precede a given ~~token~~text string and a right context automaton to populate said context record based on identified patterns that follow said given ~~token~~text string.

13. (Original) The method of claim 11 further comprising maintaining a data store of predefined token classes and assigning each token identified to at least one of said predefined token classes.

14. (Original) The method of claim 11 further comprising reporting for each token information indicative of its position, length and class membership.

15. (Original) The method of claim 11 further comprising defining a failure state and monitoring the occurrence of said failure state to maintain a record of the longest match found involving said failure state to thereby detect a default token in the absence of any matching patterns generated by said context and token automata.

16. (Original) The method of claim 11 further comprising scanning said input stream in a first direction to acquire left context information and in a second direction to acquire right context information.

17. (Original) The method of claim 11 wherein said steps of generating a context record and of segmenting the input stream collectively obey a linear time operating constraint.

18. (Currently Amended) The method of claim 11 further comprising generating tokenization information about said input stream that includes class membership of said ~~predefined~~ tokens and supplying said tokenization information to a text-to-speech synthesizer.

19. (Currently Amended) The method of claim 11 further comprising generating tokenization information about said input stream that includes class membership of said ~~predefined~~ tokens and supplying said tokenization information to a text processor.